| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Introspector.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/beans/IntrospectionException.html)   [**NEXT CLASS**](http://docs.google.com/java/beans/MethodDescriptor.html) | [**FRAMES**](http://docs.google.com/index.html?java/beans/Introspector.html)    [**NO FRAMES**](http://docs.google.com/Introspector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#17dp8vu) |

## **java.beans**

Class Introspector

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.beans.Introspector**

public class **Introspector**extends [Object](http://docs.google.com/java/lang/Object.html)

The Introspector class provides a standard way for tools to learn about the properties, events, and methods supported by a target Java Bean.

For each of those three kinds of information, the Introspector will separately analyze the bean's class and superclasses looking for either explicit or implicit information and use that information to build a BeanInfo object that comprehensively describes the target bean.

For each class "Foo", explicit information may be available if there exists a corresponding "FooBeanInfo" class that provides a non-null value when queried for the information. We first look for the BeanInfo class by taking the full package-qualified name of the target bean class and appending "BeanInfo" to form a new class name. If this fails, then we take the final classname component of this name, and look for that class in each of the packages specified in the BeanInfo package search path.

Thus for a class such as "sun.xyz.OurButton" we would first look for a BeanInfo class called "sun.xyz.OurButtonBeanInfo" and if that failed we'd look in each package in the BeanInfo search path for an OurButtonBeanInfo class. With the default search path, this would mean looking for "sun.beans.infos.OurButtonBeanInfo".

If a class provides explicit BeanInfo about itself then we add that to the BeanInfo information we obtained from analyzing any derived classes, but we regard the explicit information as being definitive for the current class and its base classes, and do not proceed any further up the superclass chain.

If we don't find explicit BeanInfo on a class, we use low-level reflection to study the methods of the class and apply standard design patterns to identify property accessors, event sources, or public methods. We then proceed to analyze the class's superclass and add in the information from it (and possibly on up the superclass chain).

Because the Introspector caches BeanInfo classes for better performance, take care if you use it in an application that uses multiple class loaders. In general, when you destroy a ClassLoader that has been used to introspect classes, you should use the [Introspector.flushCaches](http://docs.google.com/java/beans/Introspector.html#flushCaches()) or [Introspector.flushFromCaches](http://docs.google.com/java/beans/Introspector.html#flushFromCaches(java.lang.Class)) method to flush all of the introspected classes out of the cache.

For more information about introspection and design patterns, please consult the [JavaBeans specification](http://java.sun.com/products/javabeans/docs/index.html).

| **Field Summary** | |
| --- | --- |
| static int | [**IGNORE\_ALL\_BEANINFO**](http://docs.google.com/java/beans/Introspector.html#IGNORE_ALL_BEANINFO) |
| static int | [**IGNORE\_IMMEDIATE\_BEANINFO**](http://docs.google.com/java/beans/Introspector.html#IGNORE_IMMEDIATE_BEANINFO) |
| static int | [**USE\_ALL\_BEANINFO**](http://docs.google.com/java/beans/Introspector.html#USE_ALL_BEANINFO) |

| **Method Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**decapitalize**](http://docs.google.com/java/beans/Introspector.html#decapitalize(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Utility method to take a string and convert it to normal Java variable name capitalization. |
| static void | [**flushCaches**](http://docs.google.com/java/beans/Introspector.html#flushCaches())()            Flush all of the Introspector's internal caches. |
| static void | [**flushFromCaches**](http://docs.google.com/java/beans/Introspector.html#flushFromCaches(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> clz)            Flush the Introspector's internal cached information for a given class. |
| static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) | [**getBeanInfo**](http://docs.google.com/java/beans/Introspector.html#getBeanInfo(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass)            Introspect on a Java Bean and learn about all its properties, exposed methods, and events. |
| static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) | [**getBeanInfo**](http://docs.google.com/java/beans/Introspector.html#getBeanInfo(java.lang.Class,%20java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass, [Class](http://docs.google.com/java/lang/Class.html)<?> stopClass)            Introspect on a Java bean and learn all about its properties, exposed methods, below a given "stop" point. |
| static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) | [**getBeanInfo**](http://docs.google.com/java/beans/Introspector.html#getBeanInfo(java.lang.Class,%20int))([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass, int flags)            Introspect on a Java bean and learn about all its properties, exposed methods, and events, subject to some control flags. |
| static [String](http://docs.google.com/java/lang/String.html)[] | [**getBeanInfoSearchPath**](http://docs.google.com/java/beans/Introspector.html#getBeanInfoSearchPath())()            Gets the list of package names that will be used for finding BeanInfo classes. |
| static void | [**setBeanInfoSearchPath**](http://docs.google.com/java/beans/Introspector.html#setBeanInfoSearchPath(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] path)            Change the list of package names that will be used for finding BeanInfo classes. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### USE\_ALL\_BEANINFO

public static final int **USE\_ALL\_BEANINFO**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.beans.Introspector.USE_ALL_BEANINFO)

### IGNORE\_IMMEDIATE\_BEANINFO

public static final int **IGNORE\_IMMEDIATE\_BEANINFO**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.beans.Introspector.IGNORE_IMMEDIATE_BEANINFO)

### IGNORE\_ALL\_BEANINFO

public static final int **IGNORE\_ALL\_BEANINFO**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.beans.Introspector.IGNORE_ALL_BEANINFO)

| **Method Detail** |
| --- |

### getBeanInfo

public static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) **getBeanInfo**([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass)  
 throws [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html)

Introspect on a Java Bean and learn about all its properties, exposed methods, and events.

If the BeanInfo class for a Java Bean has been previously Introspected then the BeanInfo class is retrieved from the BeanInfo cache.

**Parameters:**beanClass - The bean class to be analyzed. **Returns:**A BeanInfo object describing the target bean. **Throws:** [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html) - if an exception occurs during introspection.**See Also:**[flushCaches()](http://docs.google.com/java/beans/Introspector.html#flushCaches()), [flushFromCaches(java.lang.Class)](http://docs.google.com/java/beans/Introspector.html#flushFromCaches(java.lang.Class))

### getBeanInfo

public static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) **getBeanInfo**([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass,  
 int flags)  
 throws [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html)

Introspect on a Java bean and learn about all its properties, exposed methods, and events, subject to some control flags.

If the BeanInfo class for a Java Bean has been previously Introspected based on the same arguments then the BeanInfo class is retrieved from the BeanInfo cache.

**Parameters:**beanClass - The bean class to be analyzed.flags - Flags to control the introspection. If flags == USE\_ALL\_BEANINFO then we use all of the BeanInfo classes we can discover. If flags == IGNORE\_IMMEDIATE\_BEANINFO then we ignore any BeanInfo associated with the specified beanClass. If flags == IGNORE\_ALL\_BEANINFO then we ignore all BeanInfo associated with the specified beanClass or any of its parent classes. **Returns:**A BeanInfo object describing the target bean. **Throws:** [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html) - if an exception occurs during introspection.

### getBeanInfo

public static [BeanInfo](http://docs.google.com/java/beans/BeanInfo.html) **getBeanInfo**([Class](http://docs.google.com/java/lang/Class.html)<?> beanClass,  
 [Class](http://docs.google.com/java/lang/Class.html)<?> stopClass)  
 throws [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html)

Introspect on a Java bean and learn all about its properties, exposed methods, below a given "stop" point.

If the BeanInfo class for a Java Bean has been previously Introspected based on the same arguments, then the BeanInfo class is retrieved from the BeanInfo cache.

**Parameters:**beanClass - The bean class to be analyzed.stopClass - The baseclass at which to stop the analysis. Any methods/properties/events in the stopClass or in its baseclasses will be ignored in the analysis. **Throws:** [IntrospectionException](http://docs.google.com/java/beans/IntrospectionException.html) - if an exception occurs during introspection.

### decapitalize

public static [String](http://docs.google.com/java/lang/String.html) **decapitalize**([String](http://docs.google.com/java/lang/String.html) name)

Utility method to take a string and convert it to normal Java variable name capitalization. This normally means converting the first character from upper case to lower case, but in the (unusual) special case when there is more than one character and both the first and second characters are upper case, we leave it alone.

Thus "FooBah" becomes "fooBah" and "X" becomes "x", but "URL" stays as "URL".

**Parameters:**name - The string to be decapitalized. **Returns:**The decapitalized version of the string.

### getBeanInfoSearchPath

public static [String](http://docs.google.com/java/lang/String.html)[] **getBeanInfoSearchPath**()

Gets the list of package names that will be used for finding BeanInfo classes.

**Returns:**The array of package names that will be searched in order to find BeanInfo classes. The default value for this array is implementation-dependent; e.g. Sun implementation initially sets to {"sun.beans.infos"}.

### setBeanInfoSearchPath

public static void **setBeanInfoSearchPath**([String](http://docs.google.com/java/lang/String.html)[] path)

Change the list of package names that will be used for finding BeanInfo classes. The behaviour of this method is undefined if parameter path is null.

First, if there is a security manager, its checkPropertiesAccess method is called. This could result in a SecurityException.

**Parameters:**path - Array of package names. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPropertiesAccess method doesn't allow setting of system properties.**See Also:**[SecurityManager.checkPropertiesAccess()](http://docs.google.com/java/lang/SecurityManager.html#checkPropertiesAccess())

### flushCaches

public static void **flushCaches**()

Flush all of the Introspector's internal caches. This method is not normally required. It is normally only needed by advanced tools that update existing "Class" objects in-place and need to make the Introspector re-analyze existing Class objects.

### flushFromCaches

public static void **flushFromCaches**([Class](http://docs.google.com/java/lang/Class.html)<?> clz)

Flush the Introspector's internal cached information for a given class. This method is not normally required. It is normally only needed by advanced tools that update existing "Class" objects in-place and need to make the Introspector re-analyze an existing Class object. Note that only the direct state associated with the target Class object is flushed. We do not flush state for other Class objects with the same name, nor do we flush state for any related Class objects (such as subclasses), even though their state may include information indirectly obtained from the target Class object.

**Parameters:**clz - Class object to be flushed. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If the Class object is null.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Introspector.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/beans/IntrospectionException.html)   [**NEXT CLASS**](http://docs.google.com/java/beans/MethodDescriptor.html) | [**FRAMES**](http://docs.google.com/index.html?java/beans/Introspector.html)    [**NO FRAMES**](http://docs.google.com/Introspector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).